

SDLC Delivery Model

Що таке SDLC ?

Життєвий цикл розробки програмного забезпечення (SDLC) — це процес, який виконується для проекту програмного забезпечення в рамках організації програмного забезпечення. Він складається з детального плану, який описує, як розробляти, підтримувати, замінювати та змінювати або покращувати певне програмне забезпечення. Життєвий цикл визначає методологію покращення якості програмного забезпечення та загального процесу розробки.



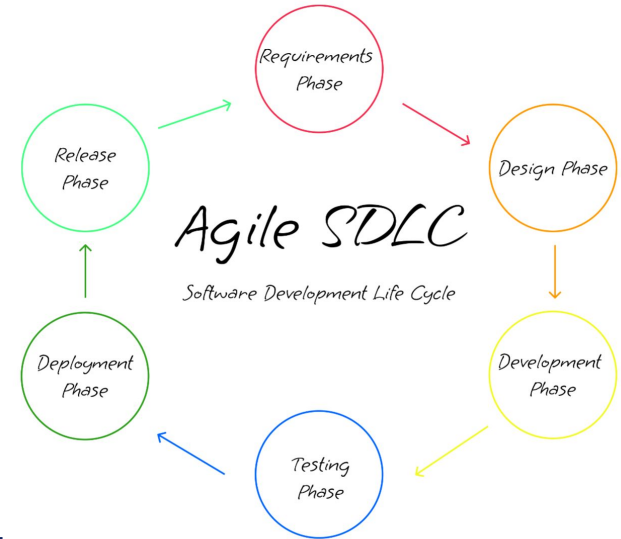
Етапи SDLC

SDLC - це систематичний процес, метою якого є створення висококласного програмного забезпечення, яке відповідає очікуванням клієнтів.

Щоб досягти цього, команда проходить 7 етапів SDLC :

- Планування;
- Аналіз вимог;
- Дизайн програмного забезпечення;
- Розробка програмного забезпечення;
- Тестування;
- Розгортання;
- Технічне обслуговування.

Ці етапи охоплюють повний життєвий цикл програмного забезпечення. Кожен етап життєвого циклу SDLC має свій процес і результати. Дотримання процесу SDLC веде до розробки програмного забезпечення систематичним і дисциплінованим способом.



Переваги та недоліки SDLC

Переваги

1. SDLC супроводжується найвищим рівнем управлінського контролю та документації.
2. Розробники розуміють, що вони повинні будувати і причини цього.
3. У всіх учасників є чіткий план досягнення конкретних цілей.
4. Усі розуміють необхідні витрати та ресурси.

Недоліки

1. Переваги SDLC існують, лише якщо команда дотримується плану.
2. Створення документації є дорогим і займає багато часу.
3. Введення користувача іноді обмежене.
4. Він не терпить змін у вимогах.



5-ий етап життєвий цикл розробки інтеграції та тестування

П'ятий етап стосується системної інтеграції, тестування програм і процедур. Це виконує інженер із забезпечення якості (QA).

Забезпечення якості — це безперервний процес, який триває до тих пір, поки програмне забезпечення не буде повністю вільним від помилок і відповідатиме всім вимогам.

Верифікація та підтвердження також є невід'ємними частинами цього етапу. Вони допомагають забезпечити успішне завершення програми.



Моделі SDLC

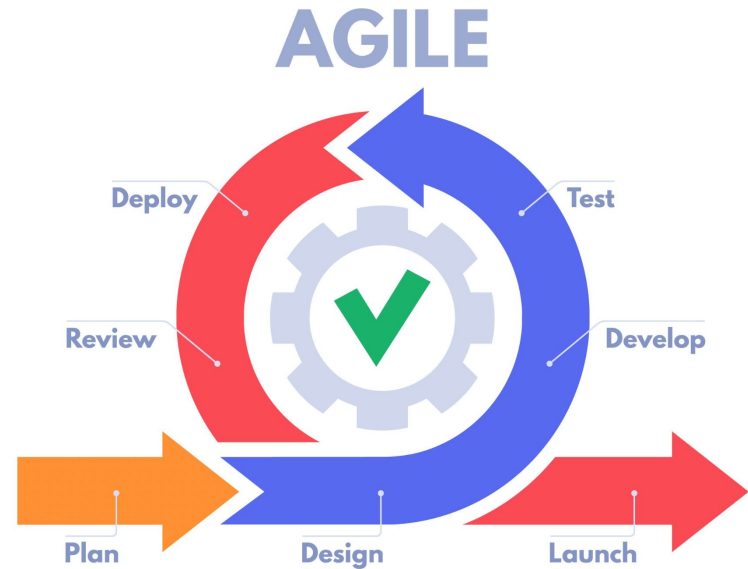
Як правило, моделі SDLC – це інструменти, які можна використовувати для кращої реалізації вашого програмного проекту. Щоб вирішити, який з них підходить для контексту проекту, ви повинні знати та розуміти кожен із них.



Agile

Підхід полягає у випуску циклів із невеликими змінами порівняно з попереднім випуском. На кожній ітерації продукт тестується. Модель Agile допомагає командам виявляти невеликі проблеми в проектах, перш ніж вони переростають у великі проблеми.

Багато команд застосовують гнучку структуру, відому як Scrum, для структурування складних проектів розробки. Scrum-команди працюють у «спринтах», які зазвичай тривають від двох до чотирьох тижнів для виконання поставлених завдань.



Плюси Agile:	Мінуси Agile:
гнучкість;	важко передбачити;
товар швидко потрапляє на ринок;	кінцевий продукт не випускається відразу;
ефективне командне спілкування.	документація залишена.

Lean

Процес Lean означає роботу над одним призначеним завданням за раз, тому немає жодних шансів на багатозадачність. Завдяки моделі Lean команди проекту можуть відмовитися від непотрібних зустрічей і скоротити обсяг документації.



Принципи Lean

Визначити цінність

Визначити цінності для клієнтів і включити ці цінності до вашого продукт

Працювати над удосконаленням

Повне виключення втрат, щоб всі дії приносили цінність клієнту через нові відкриття та безперервне покращення



Створити карту потоку цінності

Накреслити карту всіх етапів, які ведуть продукт або послугу до клієнта, включаючи ті, що несуть цінність та ті, що не несуть.

Забезпечити безперервний потік

Безперервний потік продуктів або послуг, а також інформації, від початку до кінця через процеси

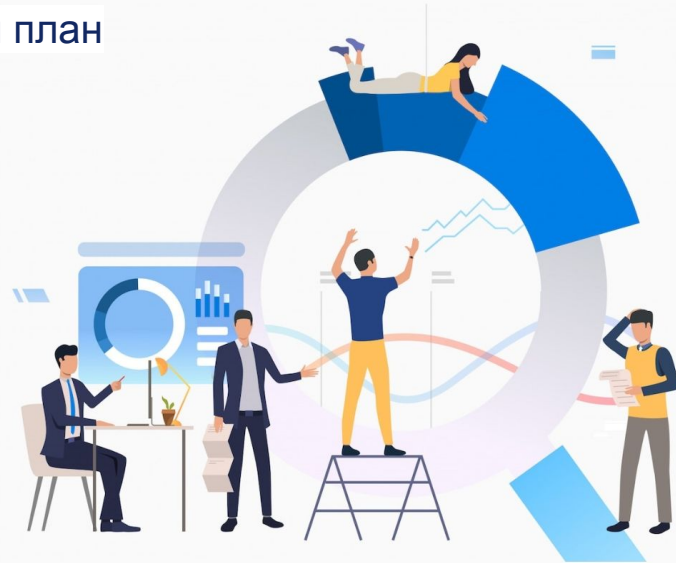
Витягування продукту

Нічого не починається на одному етапі процесу, поки це не буде необхідно на наступному етапі. Попит тягне продукт або послугу через потік цінності

Плюси Lean:	Мінуси Lean:
ефективне управління процесами;	брак часу;
розумний спосіб роботи;	відсутність стратегії;
більше уваги та цінності для клієнтів.	висока вартість реалізації.

Waterfall

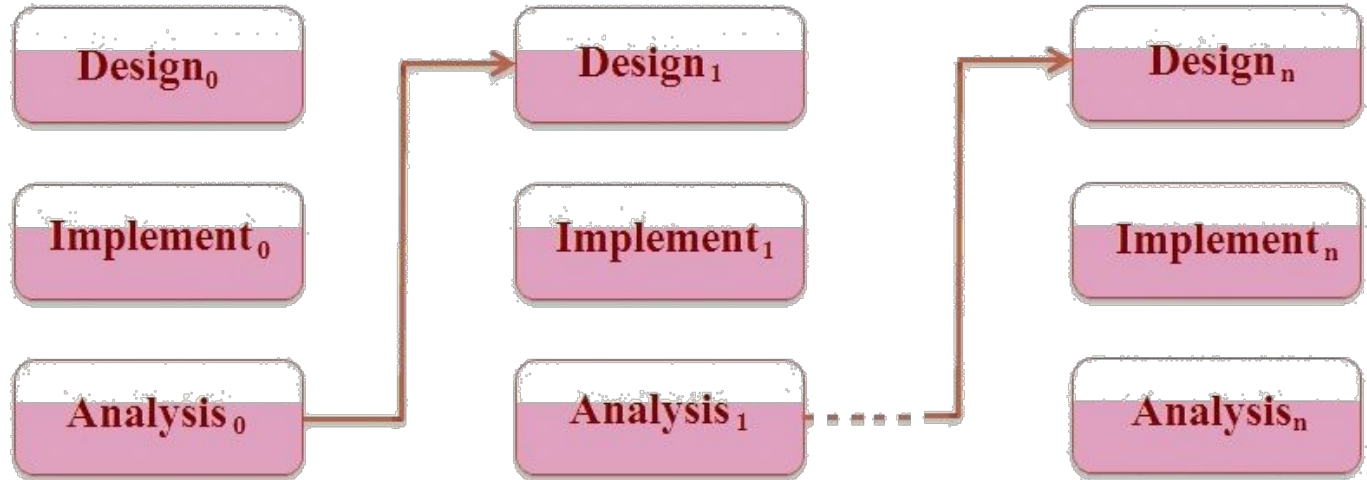
Ця модель вважається найстарішою зі структурованих методологій SDLC. Водоспад — це дуже простий підхід: закінчіть один етап, а потім перейдіть до наступного. Кожен крок базується на інформації з попереднього кроку та має свій план проекту.



Плюси Waterfall:	Мінуси Waterfall:
відсутність фінансових сюрпризів;	складне визначення потреб;
результат зрозумілий;	відсутність гнучкості;
ви отримуєте те, що плануєте;	тривалий час доставки.
часові рамки зберігаються.	

Ітеративна модель

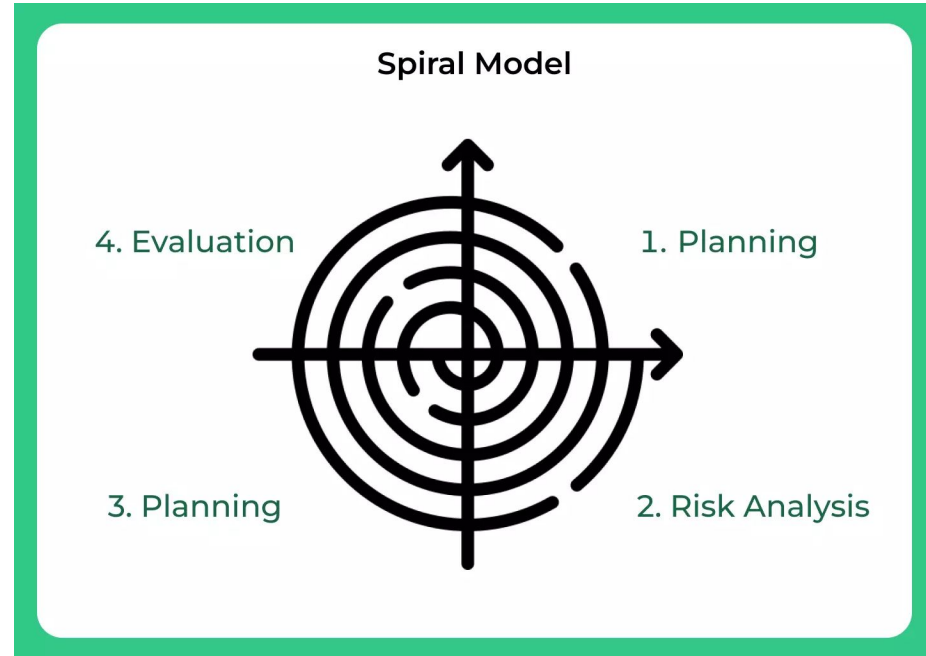
Згідно з цією моделлю, нова версія програмного забезпечення створюється з кожною фазою або ітерацією. Команда повторює ітерацію, доки вся система не буде готова.



Переваги Ітеративного:	Мінуси ітерації:
швидкий запуск проекту;	немає фіксованих бюджетів або термінів;
зниження ризику;	можливі проблеми з архітектурою.
регулярний вихід нових версій;	
ефективний зворотній зв'язок з клієнтами.	

Спіральна модель

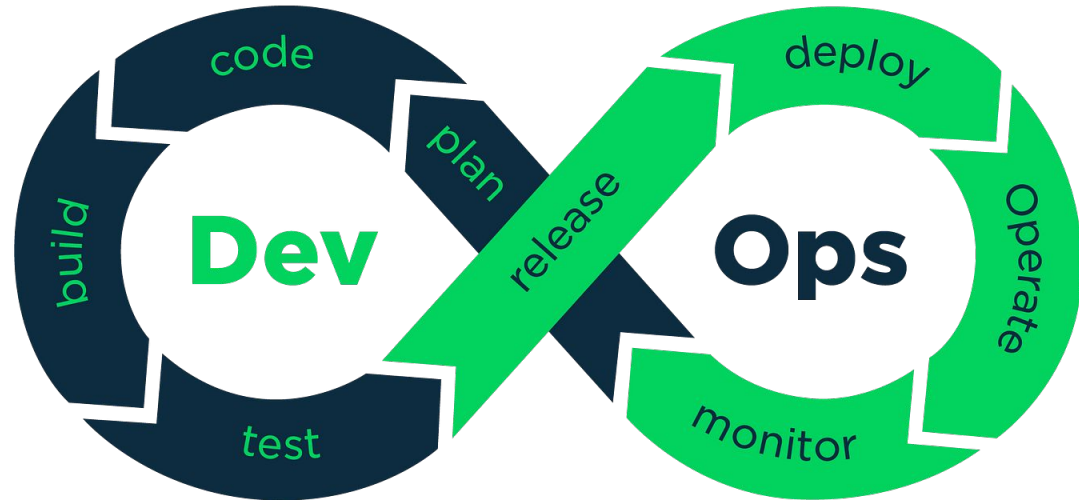
Це одна з найбільш гнучких методологій SDLC. Спіральна модель зазвичай використовується для великих проектів. Спіраль має щось спільне з ітеративною моделлю з її повторенням. Зокрема, проект проходить через 4 фази (планування, аналіз ризиків, проектування та оцінка) знову і знову по «спіралі», поки не буде завершено.



Переваги спіралі :	Мінуси спіралі:
підходить для великогабаритних виробів;	може бути дорогим у реалізації;
простий і ефективний моніторинг;	не ідеально підходить для невеликих проектів;
акцент на контроль документації.	успіх може залежати від аналізу ризиків.

DevOps model.

Ця методологія є новачком на сцені SDLC. Команди розробників і операторів тісно співпрацюють. Дисципліна, безперервний зворотний зв'язок, удосконалення процесів і автоматизація процесів розробки є відмінними рисами моделі DevOps.



Плюси DevOps:

краще управління робочим процесом;

надійні випуски;

швидке виправлення помилок.

Мінуси DevOps:

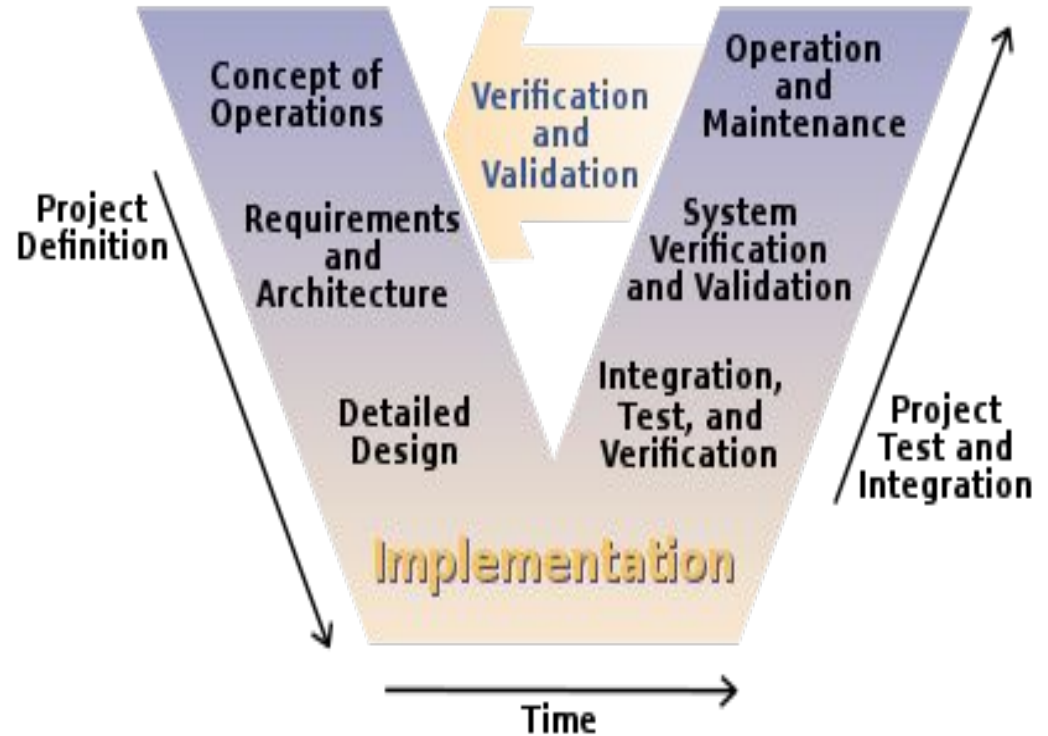
труднощі з інтеграцією;

відсутність автоматизованого тестування;

відносно високі витрати.

V Model

Модель V — це високодисциплінована модель SDLC, яка має фазу тестування, паралельну кожній фазі розробки. Модель V є розширенням моделі водоспаду, де розробка та тестування програмного забезпечення виконуються послідовним способом. Вона також відома як модель перевірки або перевірки.



Етапи розробки ПЗ - Дії, що виконуються на кожному етапі

Етап збору вимог Зберіть від клієнта якомога більше інформації про деталі та характеристики потрібного програмного забезпечення. Це не що інше, як етап збору вимог.

Стадія проектування Плануйте мову програмування, наприклад [Java](#) , [PHP](#) , .net; бази даних, як-от Oracle, [MySQL](#) тощо. Які підходять для проекту, а також деякі високорівневі функції та архітектура.

Етап побудови Після етапу проектування настає етап побудови, тобто не що інше, як кодування програмного забезпечення

Тестовий етап Далі ви тестуєте програмне забезпечення, щоб переконатися, що воно створено відповідно до специфікацій, наданих клієнтом.

Стадія розгортання Розгорніть програму у відповідному середовищі

Етап технічного обслуговування Коли ваша система буде готова до використання, вам може знадобитися змінити код пізніше відповідно до запиту клієнта